Information Sciences 672 (2024) 120661

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

AHA-3WKM: The optimization of K-means with three-way clustering and artificial hummingbird algorithm

Xiying Chen [a], Caihui Liu [a,*], Bowen Lin [a], Jianying Lai [a], Duoqian Miao [b]

[a] Department of Mathematics and Computer Science, Gannan Normal University, Ganzhou 34100, Jiangxi, China
[b] Department of Computer Science and Technology, Tongji University, Shanghai, 201804, China

ARTICLE INFO

Keywords:
K-means clustering
Artificial hummingbird algorithm
Three-way clustering
Cluster centers
Fitness function

ABSTRACT

Clustering, as an essential technique in unsupervised learning, plays a pivotal role in the fields of data mining and machine learning. However, the classic *K*-means clustering algorithm has intrinsic drawbacks such as sensitivity to initial cluster centers, susceptibility to a local optimal solution, and challenges in handling data uncertainty. To address these problems, this paper proposes an artificial hummingbird algorithm (AHA)-based three-way *K*-means clustering algorithm, called AHA-3WKM. First, AHA is introduced to address the problems of sensitivity to initial cluster centers and local optima. Second, a fitness function of AHA is specifically constructed to find the best initial clustering centers so that the hummingbirds can search for high-quality food sources, i.e., the global optimum cluster centers. Third, a three-way clustering approach is utilized to capture information about data uncertainty. In this way, the results of clustering are divided into three distinct regions based on the relationship between objects and clusters. The experimental results demonstrate that AHA-3WKM has good performance, and enhances the stability and the accuracy of clustering results.

1. Introduction

In the era of rapid development of information technology [1,2], the generation of data has shown an unprecedented speed and scale. Consequently, the extraction of meaningful information from massive data has become a crucial challenge in the fields of data mining and machine learning. Clustering, as an important unsupervised learning technique, has been widely adopted in the analysis of massive data [3]. The primary goal of clustering is to categorize unlabeled data into different clusters, so that data points within the same cluster demonstrate a high degree of similarity, while the similarity in different clusters is low. Clustering has a wide range of applications in different domains, including image processing [4], bioinformatics [5], recommendation systems [6], air transportation [7], and so on.

The classic *K*-means algorithm is one of the most commonly used clustering methods that is simple and effective. Its fundamental principle involves dividing data into *K* clusters by minimizing the sum of distances between each data point and the center of its respective cluster. Although the *K*-means has been broadly applied in the variety of fields owing to its simplicity and efficiency [8], there are still the following problems that need to be improved.

* Corresponding author.
E-mail addresses: xiying_chen@163.com (X. Chen), liucaihui@gnnu.edu.cn, liu_caihui@163.com (C. Liu), lin_bw@qq.com (B. Lin), 1421631021@qq.com (J. Lai), dqmiao@tongji.edu.cn (D. Miao).

https://doi.org/10.1016/j.ins.2024.120661
Received 15 November 2023; Received in revised form 24 March 2024; Accepted 24 April 2024

Available online 30 April 2024

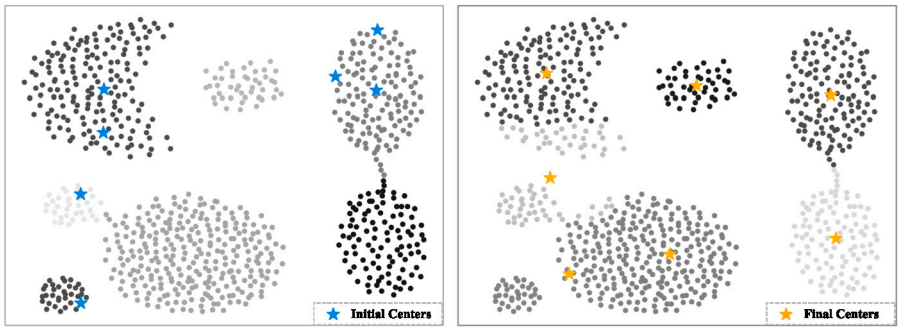


Fig. 1. An example on the Aggregation dataset, which can demonstrate the sensitivity of *K*-means algorithm to initial cluster centers.

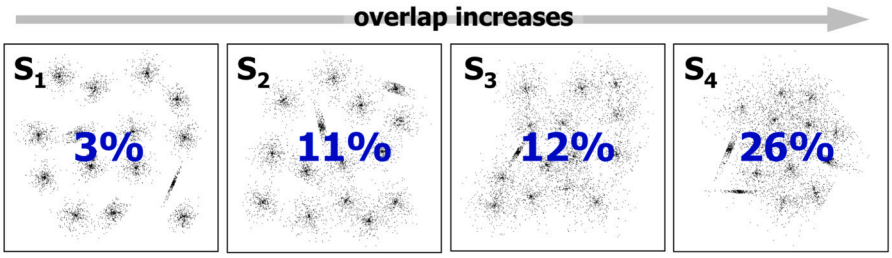


Fig. 2. The accuracy(%) of the *K*-means algorithm increases as the overlap between different clusters increases [11].

(1) Sensitivity to initial cluster centers. The results of *K*-means algorithm are significantly influenced by the initial cluster centers selected. Fig. 1 illustrates the sensitivity of *K*-means algorithm to initial cluster centers.

(2) Proneness to local optima. Another challenge is the algorithm's susceptibility to getting trapped in local optima. Inappropriate initial center selection may cause the algorithm's iterations to converge to local optima, resulting in unstable or inaccurate clustering results [9].

(3) Difficulty in capturing the relationships among uncertain information. *K*-means is a hard clustering algorithm with clear-cut cluster boundaries. It performs better with more overlapping clusters [10], as shown in Fig. 2. However, the more the overlapping clusters, the higher likelihood that data points in these regions might exhibit similarities with multiple clusters. This further hinders the *K*-means algorithm from effectively capturing relationships among uncertain information, thereby limiting its ability to improve the clustering accuracy and increasing the risk in decision-making.

To address the problems of the *K*-means algorithm mentioned in (1) and (2), many scholars have attempted to enhance it by using Swarm Intelligence (SI). SI provides an innovative problem-solving approach by simulating the principles of natural selection observed in biological systems, which can effectively find the global optimal solutions to given problems [12]. Saida et al. [13] proposed a *K*-means clustering algorithm based on the cuckoo search algorithm. This algorithm employs the Sum of Squared Errors (SSE) as the fitness function to find the optimal cluster centers. Wang et al. [14] improved the *K*-means algorithm by using the flower pollination algorithm with bee pollinators. The algorithm defines the fitness function by minimizing the distance between each data point and the center of its respective cluster. Nayak et al. introduced Particle Swarm Optimization (PSO) [15] and Firefly Algorithm(FA) [16] to enhance *K*-means. Both PSO and FA included two self-defined parameters, k and d, in the fitness function to calculate the sum of the distances between objects and cluster centers. Li et al. [17] proposed an algorithm, called FPSO-GAK, based on fuzzy system, PSO, and genetic algorithm (GA). They optimized the parameters of PSO using a fuzzy system and further refined the results using GA. The sum of distances between objects and cluster centers serves as the fitness function of the algorithm. Despite these methods having shown powerful ability, their applications are still limited by two challenges. The first challenge involves the construction of the fitness function, which mainly focuses on the intra-cluster similarity while ignoring the inter-cluster similarity, or although both aspects are considered, the defined formula is computationally complex which may reduce the efficiency of the algorithm. In addition, the inherent challenge of the *K*-means algorithm still exists, that is, it cannot effectively capture the uncertain information within the dataset.

In this paper, we introduce the Artificial Hummingbird Algorithm (AHA) to address the problems of the sensitivity to initialization and the proneness to local optima. AHA [18] is a swarm intelligence algorithm, which mimics the flight and foraging behaviors of hummingbirds. It is used as a meta-heuristic approach for global optimization problems. AHA has shown excellent performance in addressing problems across low to medium dimensions and has potential adaptability to diverse search space magnitudes [19]. Thus, it has robustness in finding the optimal solutions. These characteristics collectively make AHA well-suited for addressing the above two problems of *K*-means clustering. In addition, AHA is widely used because of its simplicity in implementation and ability to produce high-quality solutions with minimal parameter tuning. For example in the domains of artificial intelligence [20], engineering technology [21], clean energy [22]. We define the fitness function of AHA based on the clustering concepts of high intra-cluster similarity and low inter-cluster similarity, and the definition is simplified to improve the computational efficiency. This fitness

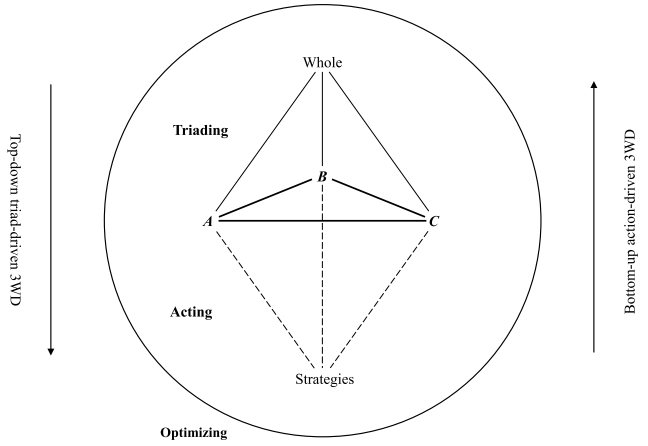


Fig. 3. The TAO (Triading-Acting-Optimizing) framework of three-way decision [23].

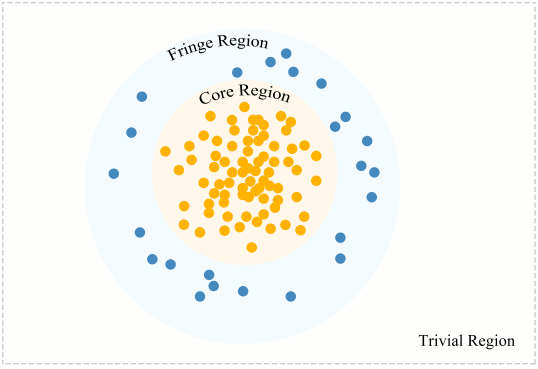


Fig. 4. Diagram of the three-way clustering.

function enables the hummingbird swarm to purposely optimize clustering outcomes, attains the optimal initial cluster centers, and speeds up the convergence of the algorithm. Thus, this approach overcomes the problem of local optima and improves the efficiency of clustering.

To solve problem (3), this paper incorporates the three-way decision into clustering analysis. The three-way decision theory was introduced by Yao [24,25] as a strategy to tackle the problem of information uncertainty. The fundamental idea involves constructing the universe into a triad and adopting a set of strategies for processing the triad. The framework of three-way decision can be described as the TAO (Triading-Acting-Optimizing), as shown in Fig. 3. Yu [26] applied the concept of three-way decision to clustering analysis and proposed the method of three-way clustering. This innovative method incorporates the concept of fringe regions into traditional binary clustering results and establishes three regions within each cluster, which effectively captures the uncertainty of clustering and reduces the risk of decision-making associated with inaccuracy. The three regions within each cluster are the core region, the trivial region, and the fringe region, as depicted in Fig. 4. These regions represent three distinct types of relationships between objects and clusters: (1) the objects within the core region are definitively assigned to a cluster; (2) the objects within the trivial region are definitively excluded from a cluster; (3) the objects within the fringe region may belong to a cluster. By incorporating the fringe region, the three-way clustering quantifies the extent of the influence that samples have on their respective clusters, thereby solving the problem of uncertainty in traditional clustering methods and reducing the decision risks caused by uncertain information.

Recently, scholars have endeavored to integrate the three-way clustering theory into K-means clustering, and promising results have been yielded. Wang et al. [27] introduced a novel three-way K-means clustering algorithm (called TWKM). TWKM incorporates the overlapping clusters and utilizes the perturbation analysis to implement the three-way clustering. However, similar to the conventional K-means algorithm, the three-way K-means algorithm is also influenced by the initial selection of cluster centers, which makes it prone to local optima. In response to this challenge, some scholars have integrated swarm intelligence algorithms into this context for improvement. Gao et al. [28]proposed a method that combines PSO with the three-way K-means algorithm. In this method, SSE serves as the fitness function of PSO to optimize the cluster centers, and the difference sorting method of three-way clustering is used to measure the relationships between data points and clusters. Similarly, Guo et al. [29] introduced the Ant Colony Optimization (ACO) into the three-way K-means clustering process. They used the ACO to address the susceptibility of K-means to local optima. Each of the aforementioned algorithms has considered both intra-cluster and inter-cluster similarities when defining

the fitness function and has been verified for its efficiency through experimentation. It is worth noting that, currently, there are relatively few methods that combine swarm intelligence with the three-way clustering to enhance the *K*-means algorithm.

To address the aforementioned issues, this paper proposes an artificial hummingbird algorithm-based three-way *K*-means clustering algorithm, called AHA-3WKM. AHA-3WKM uses the flight patterns and foraging strategies of hummingbirds through multiple iterations to discover the optimal food source locations, which are then utilized as the optimal initial cluster centers. Subsequently, the three-way decision framework is applied to handle the results of *K*-means clustering, and further divides the clustering results into core regions and fringe regions based on the similarity between data points and clusters. Experimental comparisons are conducted on 14 UCI datasets against 6 relevant clustering algorithms, and the experiment results show that AHA-3WKM exhibits higher accuracy. In addition, the results in terms of Davies-Bouldin Index (DBI) demonstrate that clustering results generated by the proposed algorithm have low inter-cluster similarity and high intra-cluster similarity. Our approach effectively alleviates the problems related to the randomness of cluster centers and the inclination of clustering to converge to local optima. Moreover, it categorizes each cluster into a core region and a fringe region, thereby enhancing its ability to represent the uncertainty within the dataset and reducing the risk of decision-making.

The main contributions of this paper are as follows.

(1) AHA is introduced to address the problems of the sensitivity to initial cluster centers and the proneness to local optima. Hummingbirds are treated as data points, which dynamically update their strategies and effectively find the optimal cluster centers during multiple iterations.

(2) A fitness function is designed based on the clustering principle of "birds of a feather flock together", with the aim of simplifying calculations, which enhances the specificity and practicality of *K*-means algorithm.

(3) An AHA-based three-way *K*-means clustering algorithm (i.e., AHA-3WKM) is proposed. The clustering process is initialized with cluster centers optimized by AHA, and the results are represented in three regions, which can capture the uncertainty within the datasets.

(4) Comparative experiments are conducted on 14 UCI datasets, and the results demonstrate the effectiveness of the proposed algorithm.

The remaining sections of this paper are organized as follows. Section 2 introduces the fundamental concepts of the three-way clustering and AHA. In Section 3, we delve into the intricate details of AHA-3WKM. Section 4 is dedicated to verifying the effectiveness of the proposed algorithm through experiments. Finally, the conclusions and the outlines of potential future work are given in Section 5.

2. Preliminary

In this section, we provide an overview of concepts related to our algorithm, including those in three-way clustering, three-way *K*-means, and AHA.

2.1. Three-way clustering

Assume that $U = \{x_1, x_2, ..., x_n, ..., x_N\}$ is a nonempty finite set of objects called the universe, and $CS = \{C_1, C_2, ..., C_K, ..., C_K\}$ is a family of K clusters, where for any $1 \le k \le K$, $C_k \in U$, and $U = \bigcup_{k=1}^{K} C_k$. Differing from hard clustering, which represents each cluster as a single set, three-way clustering divides each cluster $C_k (1 \le k \le K)$ into three regions: the core region (Co), the fringe region (Fr), and the trivial region (Tr). For any $x \in U$, if $x \in Co(C_k)$, then the object x definitely belongs to the cluster C_k. If $x \in Fr(C_k)$, then it implies that x could potentially be associated with C_k. If $x \in Tr(C_k)$, then x definitely does not belong to C_k.

Typically, the three regions of $C_k (1 \le k \le K)$ satisfy the following properties:

(1) $Tr(C_k) = U - Co(C_k) - Fr(C_k)$

(2) $Co(C_k) \cap Fr(C_k) = \varnothing$

(3) $Co(C_k) \cap Tr(C_k) = \varnothing$

(4) $Tr(C_k) \cap Fr(C_k) = \varnothing$

Based on the aforementioned discussion, each cluster $C_k (1 \le k \le K)$ in the three-way clustering can be represented as a pair of sets:

$$C_k = (Co(C_k), Fr(C_k)),$$

where for any $1 \le k \le K$, $Co(C_k)$ denotes the core region of the cluster C_k, and $Fr(C_k)$ denotes the fringe region of C_k.

In general, the core region and the fringe region satisfy the following properties:

(i) $\forall C_k \in CS,\ Co(C_k) \neq \varnothing,$

(ii) $\bigcup_{k=1}^{K} (Co(C_k) \cup Fr(C_k)) = U,$

(iii) $Co(C_k) \cap Co(C_j) = \varnothing,\ j \neq k.$

Property (i) ensures that each cluster has a non-empty core region, indicating the presence of at least one object in each cluster. Property (ii) ensures that we can effectively partition all objects through clustering. Property (iii) ensures that the core regions of different clusters are mutually exclusive. Based on the discussion above, the result of three-way clustering can be represented as:

$$CS = \{(Co(C_1), Fr(C_1)), (Co(C_2), Fr(C_2)), ..., (Co(C_k), Fr(C_k)), ..., (Co(C_K), Fr(C_K))\}.$$

It is evident that, in the case of $Fr(C_k) = \emptyset$, three-way clustering generates the same results as hard clustering, where C_k is represented as $Co(C_k)$ and $Tr(C_k) = U - Co(C_k)$. It can be seen that hard clustering is a special case of three-way clustering, and the latter provides an effective solution for handling data uncertainty.

2.2. Three-way K-means

The Three-way K-means clustering algorithm incorporates the three-way decision into the standard K-means clustering. This algorithm mainly consists of the following two steps.

Step 1: It employs the overlapping clustering method to obtain the support set by determining the minimum distance between an object and its nearest cluster center. For each object v, if the difference between the distance from v to other cluster centers and the minimum distance is less than α (α is a given threshold), then v is assigned to the supports.

Assume that $U = \{x_1, x_2, \dots, x_n, \dots, x_N\}$ is a nonempty finite set of objects (called the universe), and the results of three-way K-means clustering are represented as $CS = \{(Co(C_1), Fr(C_1)), (Co(C_2), Fr(C_2)), \dots, (Co(C_K), Fr(C_K))\}$, where for each cluster $C_k (1 \le k \le K)$, the support P_k is the union of the core region and the fringe region, i.e.,

$$P_k = Co(C_k) \cup Fr(C_k),\ 1 \le k \le K. \tag{1}$$

Assume that v is an object, and $c_1, c_2, ..., c_K$ are K initial cluster centers which are randomly selected from U. By calculating the minimum distance from object v to the K cluster centers, i.e., $d(v, c_i) = \min_{1 \le k \le K}(v, c_k)$, we can obtain the set $S = \{j : d(v, c_j) - d(v, c_i) \le \alpha \ and \ i \ne j\}$, where α is a pre-defined parameter. This process may lead to two scenarios:

(1) $if\ S = \emptyset,\ then\ v \in P_i$

(2) $if\ S \ne \emptyset,\ then\ v \in P_i\ and\ v \in P_j$

Subsequently, the center of each cluster is updated using the following equation:

$$c_k = \frac{\sum_{v \in P_k} v}{|P_k|},\ 1 \le k \le K, \tag{2}$$

where $v \in P_k$ is an object in P_k, and $|P_k|$ denotes the cardinality of set P_k.

Step 2: It uses perturbation analysis to partition the supports into the core region and the fringe region. This approach effectively categorizes the elements within $P_k (1 \le k \le K)$ into two specific categories.

$$\text{condition I} = \{v \in P_k \mid \exists j = 1, 2, \dots, K,\ j \ne k,\ v \in P_j\};$$

$$\text{condition II} = \{v \in P_k \mid \forall j = 1, 2, \dots, K,\ j \ne k,\ v \notin P_j\}.$$

When an object v satisfies condition I, it indicates that v exists in multiple supports. In this case, $v \in Fr(C_k)$. Condition II indicates that the object v only exists in one support. In this scenario, multiple identical objects are added to P_k, forming the new support P_k^*. Subsequently, the new center c_k^* is computed using Equation (2), and the distance between the new and old centers is then compared. If this distance is less than a given parameter β, then $v \in Co(C_k)$, otherwise $v \in Fr(C_k)$.

2.3. Artificial hummingbird algorithm (AHA)

AHA is a bio-inspired optimization algorithm based on the special flight and intelligent foraging behaviors of hummingbirds. According to the comparative research in [18], this algorithm has strong global search capability and high optimization accuracy. The algorithm constructs a visit table to keep track of each hummingbird's visitation level to every food source. And the visit table indicates the time since the last visit to a particular food source and its fitness. Each food source acts as a solution vector and participates in optimization operations, with its quality determined by the value of fitness function. The position of each hummingbird corresponds to the location of a food source, and the fitness value represents the nectar volume of the food source (the higher the fitness value, the greater the nectar volume). The foraging behaviors of hummingbirds consist of three stages: guided foraging, territorial foraging, and migration foraging.

2.3.1. Initialization

Assume that a population of n hummingbirds is randomly placed on n food sources, let x_i represent the location of the ith food source, which is randomly initialized as follows:

$$h_i = Low + r \times (Up - Low)\ ,\ i = 1, 2, ..., n, \tag{3}$$

where Low and Up represent the lower and upper boundaries of the d-dimensional problem, respectively, and r is a random vector in $[0, 1]$. Each hummingbird can locate a target food source using the visit table. The visit table is typically updated during each iteration, and is initialized as follows:

$$VT_{i,j} = \begin{cases} 0 & if\ i \neq j \\ null & if\ i = j \end{cases},\ i, j = 1, 2, ..., n, \tag{4}$$

where i represents the ith hummingbird, j represents the jth food source. If $i = j$, then $VT_{i,j} = null$, indicating that the ith hummingbird is feeding on its own food source. If $i \neq j$, then $VT_{i,j} = 0$, indicating that during the current iteration, the ith hummingbird has just visited the jth food source, which represents the visitation level of the ith hummingbird to the jth food source.

2.3.2. Flight patterns

AHA defines three flight patterns of hummingbirds: axial flight, diagonal flight, and omnidirectional flight. During the foraging process, the flying skills of hummingbirds are simulated by introducing direction switch vectors, which are used to control their direction in the d-dimensional space. The axial flight is defined as follows:

$$V^{(i)} = \begin{cases} 1, & if\ i = rand([1, d]) \\ 0, & otherwise \end{cases},\ i = 1, 2, ..., d. \tag{5}$$

The diagonal flight is defined as follows:

$$V^{(i)} = \begin{cases} 1, & if\ i = S(j), j \in [1, k], S = permrand(k), k \in [2, \lceil r_1 \times (d-2) \rceil + 1] \\ 0, & otherwise \end{cases},\ i = 1, 2, ..., d. \tag{6}$$

The omnidirectional flight is defined as follows:

$$V^{(i)} = 1\ ,\ i = 1, ..., d, \tag{7}$$

where $rand([1, d])$ generates a random integer between $[1, d]$, $permrand(k)$ generates a random integer between $[1, k]$, and r_1 represents a random number within $[0, 1]$. The axial flight means that hummingbirds can fly along any coordinate axis; the diagonal flight allows hummingbirds to move from one corner of a rectangle to another, determined by any two axes in the coordinate system; the omnidirectional flight means that any flight direction can be projected onto every individual coordinate axis. Only hummingbirds are capable of axial and diagonal flights.

2.3.3. Guided foraging

The guided foraging is an optimization strategy in which hummingbirds search for food sources near the location of the highest visitation level. According to the visit table, the algorithm prioritizes the visit to the food source that has not been visited for the longest time. In cases where visitation levels are equal, the food source with the highest nectar-refilling rate is generally favored by hummingbirds.

The updated location of a new food source is determined based on both the flight direction of the hummingbird and its target food source, which is defined as follows:

$$h_i^{new}(t+1) = h_{i,tar}(t) + a \times V \times (h_i(t) - h_{i,tar}(t)), \tag{8}$$

where $h_i^{new}(t+1)$ represents the updated food source location of the ith hummingbird; $h_i(t)$ denotes the position of the ith hummingbird at the tth iteration; $h_{i,tar}(t)$ denotes the target food source location with the highest visitation level for the ith hummingbird; a represents the guided factor following a normal distribution, i.e., $a \sim N(0, 1)$; and V is the direction change vector of hummingbirds, calculated by Equations (5)-(7). Moreover, $h_i^{new}(t+1)$, $h_{i,tar}(t)$, $h_i(t)$ and V are all d-dimensional vectors.

After updating the location, the nectar-refilling rate of the new food source location is revised, and the subsequent foraging location of the hummingbird is determined by comparing it with the current location, which is defined as follows:

$$h_i(t+1) = \begin{cases} h_i(t) & f(h_i(t)) \leq f(h_i^{new}(t+1)) \\ h_i^{new}(t+1) & f(h_i(t)) > f(h_i^{new}(t+1)) \end{cases}. \tag{9}$$

2.3.4. Territorial foraging

The territorial foraging is another optimization strategy employed by AHA, enabling hummingbirds to navigate towards their own food source locations. This strategy is updated based on Equation (10).

$$h_i^{new}(t+1) = h_i(t) + b \times V \times h_i(t), \tag{10}$$

where $h_i^{new}(t+1)$ represents the updated location of the food source for the ith hummingbird; $h_i(t)$ denotes the position of the ith hummingbird at the tth iteration; b represents the territorial factor following a normal distribution, i.e., $b \sim N(0, 1)$; and V is a d-dimensional vector representing the direction change of hummingbirds.

After discovering a new food source, the position update of hummingbirds is performed according to the foraging update rule outlined in Equation (9).

2.3.5. Migration foraging

The migration foraging is a crucial step in the later stages of the algorithm to prevent falling into local optima. When hummingbirds encounter food shortages in the regions they visit, they migrate to more distant regions in search of new food sources. Once the number of iterations reaches a predetermined value based on the migration coefficient ($t = 2n$, where n is the population size), the hummingbird with the lowest fitness abandons its current food source. It then conducts a random search in the search space for a new food source and updates its position accordingly, which is defined as follows:

$$h_{wor}(t+1) = Low + c \times (Up - Low), \tag{11}$$

where $h_{wor}(t+1)$ is the updated location of the food source with the worst fitness; c is a d-dimensional vector of random numbers within $[0, 1]$; and Up and Low denote the upper and lower boundaries of the optimization problem, respectively.

2.3.6. Visit table updates

The visit table is automatically updated after the completion of any foraging model. After the guided foraging, the visitation level $h_{i,tar}(t)$ of the current hummingbird's target food source is reset to zero, while the visitation levels of other food sources are increased by one unit. In cases of territorial foraging or migration foraging, the visitation level $h_{i,tar}(t)$ is increased by one unit. In addition, if the newly generated position of a hummingbird after foraging can lead to an improved fitness, then the visitation levels of that food source are increased by one unit based on their respective highest visitation level. Furthermore, the visitation levels of all hummingbirds will be updated by increasing them by one unit if a hummingbird discovers a new foraging position with higher fitness, based on their respective highest visitation levels.

3. The proposed algorithm

In this section, we introduce the relevant concepts of the proposed algorithm and provide the steps involved. To overcome the randomness of initial cluster centers and the problem of clustering being trapped in local optima, we introduce AHA and define a corresponding fitness function to find the optimal food source locations as the optimal initial cluster centers. Next, we use the obtained cluster centers for clustering and employ the three-way clustering method to partition the clustering results into three regions, aiming to better capture the uncertainty in objects.

3.1. Basic concepts of AHA-3WKM

The fitness function plays a crucial role in swarm intelligence algorithms, guiding the search and optimization process of SI and ultimately determining the quality of the solution found. In order to ensure efficient exploration and identification of high-quality solutions in clustering problems, the fitness function of AHA in this paper is defined based on the fundamental principles of maximizing the intra-cluster similarity and minimizing the inter-cluster similarity. The high-quality clustering results typically exhibit high intra-cluster similarity and low inter-cluster similarity [30]. In AHA, the nectar-refilling rate of food sources has a significant contribution to the formation of hummingbirds' foraging behaviors, and is quantified as a fitness value calculated by the fitness function. The nectar-refilling rate of food sources increases with the fitness value.

Definition 1 *(Intra-cluster Cohesion of A Cluster).* Let $U = \{x_1, x_2, \ldots, x_n, \ldots, x_N\}$ be a nonempty finite set of objects, $CS = \{C_1, C_2, \ldots C_k, \ldots, C_K\}$ be a partition of U, and $C = \{c_1, c_2, \ldots, c_k, \ldots, c_K\}$ be the set of cluster centers. For each cluster $C_k \in CS (1 \leq k \leq K)$, the intra-cluster cohesion $Cohe(C_k)$ of C_k is defined as:

$$Cohe(C_k) = \frac{1}{|C_k|} \sum_{x_i \in C_k} dist(x_i, c_k), \tag{12}$$

where for any $c_k \in C$ and $x_i \in C_k$, $dist(x_i, c_k)$ denotes the distance between object x_i and cluster center c_k, and $|C_k|$ represents the cardinality of set C_k.

In order to comprehensively represent the clustering results, we take the minimum cohesion value of all clusters as the intra-cluster cohesion metric of CS (denoted as $Cohe(CS)$), that is,

$$Cohes(CS) = \min_{k=1,2,\ldots,K} \{Cohe(C_k)\}. \tag{13}$$

$Cohes(CS)$ is used to assess the overall compactness of clustering results obtained at different numbers of clusters. The larger the value of $Cohes(CS)$, the better the cohesion of the entire clustering result, while the smaller the value, the worse the cohesion.

Definition 2 *(Inter-cluster Separation Metric).* Let $U = \{x_1, x_2, \ldots, x_n, \ldots, x_N\}$ be a nonempty finite set of objects, $CS = \{C_1, C_2, \ldots C_k, \ldots, C_K\}$ be a partition of U, and $C = \{c_1, c_2, \ldots, c_k, \ldots, c_K\}$ be the set of cluster centers. The inter-cluster separation metric of CS (denoted as $Sepcs(CS)$) is defined as:

$$Sepcs(CS) = \max_{i \neq j} dist(c_i, c_j), \tag{14}$$

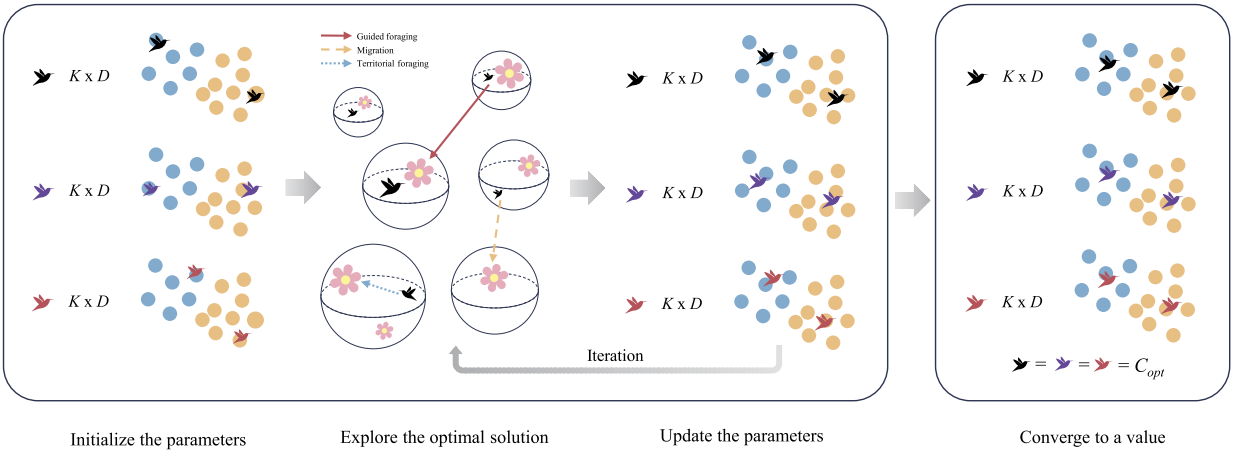


Fig. 5. Searching process of the optimal cluster centers C_{opt}.

where for any $c_i,\ c_j \in C(i \neq j)$, $dist(c_i, c_j)$ denotes the distance between cluster centers c_i and c_j.

$Sepcs(CS)$ is used to comprehensively assess the separability of clustering results under different numbers of clusters. The higher the value of $Sepcs(CS)$, the higher the degree of separation in the entire clustering result, while the lower the value, the lower the degree of separation.

Due to the difference in scale between $Cohes(CS)$ and $Sepcs(CS)$, the intra-cluster cohesion metric is normalized during computation, which is defined as follows:

$$Cohes(CS) = \frac{\max\limits_{k=1,2,\cdots,K}\{Cohe(C_k)\}}{\sum\limits_{k=1}^{K} Cohe(C_k)}. \tag{15}$$

Definition 3. The fitness function of AHA is defined as follows.

$$fit_{AHA}(CS) = Cohes(CS) - Sepcs(CS), \tag{16}$$

where $Cohes(CS)$ is the intra-cluster cohesion metric of CS, and $Sepcs(CS)$ is the inter-cluster separation metric of CS.

The fitness value increases when $Cohes(CS)$ increases and $Sepcs(CS)$ decreases, indicating a better quality of clustering with closer objects within each cluster and greater separation between different clusters. On the contrary, if the cohesion within each cluster is low and the separation between different clusters is high, then the fitness value will be low, indicating a poorer clustering quality. By maximizing this fitness function, we can find the food source with the optimal nectar-refilling rate as an approximate global optimum value, which represents the cluster center with the best quality.

Definition 4 *(Optimal Cluster Centers).* Let $U = \{x_1, x_2, \ldots, x_n, \ldots, x_N\}$ be a nonempty finite set of objects, and $CS = \{C_1, C_2, \ldots C_k, \ldots, C_K\}$ be a partition of U. The optimal cluster centers C_{opt} obtained by AHA is defined as:

$$C_{opt} = \arg\max_{CS} fit_{AHA}(CS). \tag{17}$$

To better understand the acquisition of C_{opt}, we briefly outline the optimization process in Fig. 5. This process comprises four key steps: initialization, optimization, updating, and multiple iterations until convergence. The hummingbird swarm is regarded as a set of cluster centers with $K \times D$ dimensions, where K is the number of clusters, and D is the dimension of data. In Fig. 5, we set $nPop = 3$, $K = 2$, and $D = 2$, where $nPop$ is the count of hummingbird. First, during the initial phase, we randomly select K data points from the dataset as the initial cluster centers. Second, we perform the optimization process based on fitness values and visitation levels to maximize the fitness function. Third, all parameters are updated, and the convergence state is reached by iterating over two processes (i.e., the optimization process and the updating process). Finally, all hummingbirds converge to the same position, i.e., the solution for the optimal cluster centers C_{opt}.

The following example demonstrates how hummingbirds perform a guided foraging strategy in the first iteration to find the target food source and explore the optimal solutions. As shown in Fig. 6, $f(h)$ is the fitness value of the current position, and $f(h^{new})$ is the updated fitness value. Given three hummingbirds, their positions and the visit tables are initialized using Equations (2) and (3), respectively.

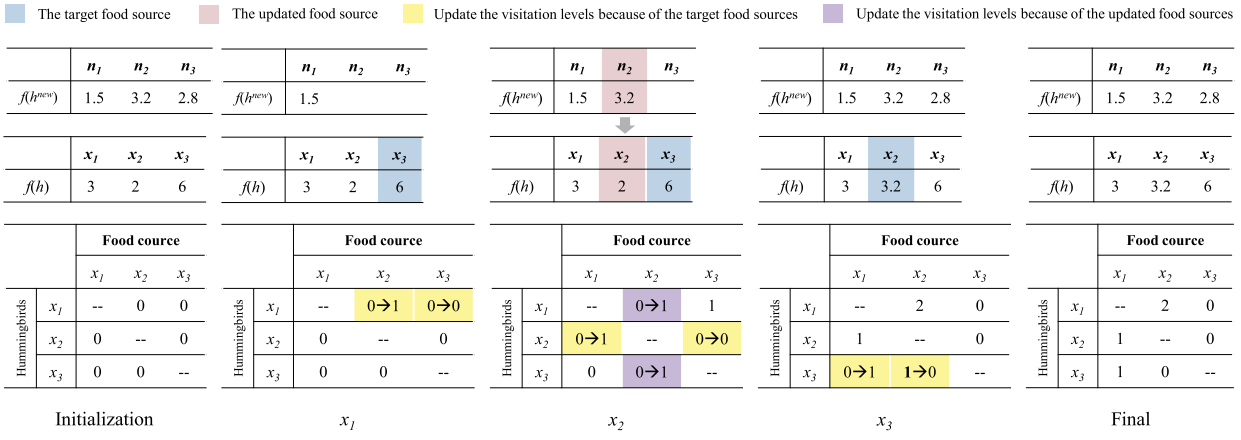


Fig. 6. Example for exploring the optimal solutions with a guided foraging strategy in the first iteration.

The first hummingbird x_1 finds two food sources: x_2 and x_3, which have the same visitation level, i.e., 0. Since the fitness value of x_3 is higher than that of x_2, x_3 is the target food source of x_1. Then, Equations (8) and (9) are applied to update the visitation levels. In this process, the target food source x_3 is initialized to 0, and the food source x_2 is increased by 1 because it is not visited.

The second hummingbird x_2 also finds two food sources: x_1 and x_2, which have the same visitation level, i.e., 0. Since the fitness value of x_3 is higher than that of x_1, x_3 is the target food source of x_2. Then, Equations (8) and (9) are applied to update the visitation levels. In this process, the target food source x_3 is initialized to 0, and the food source x_1 is increased by 1. In addition, since the updated fitness value n_2 is better than the food source x_2, x_2 is replaced with n_2, and the visitation level of x_2 to other hummingbirds needs to be increased by 1 based on their corresponding row's highest visitation level.

For the third hummingbird x_3, the food source of x_2 is the target food source since it has the highest visitation level. Therefore, the visitation level of the target food source x_2 is initialized to 0, and the visitation levels of the food sources of x_1 are increased by 1.

After one iteration, the updated visit table is shown as 'Final' in Fig. 6. In this process, the positions of all hummingbirds are updated.

Based on Definition 4, we introduce the process of three-way K-means clustering (3WKM). The 3WKM process mainly consists of two steps: firstly, utilizing overlapping clustering method to obtain the upper approximate support sets (the supports) for clustering; secondly, employing perturbation analysis to divide the upper approximate support sets into core region and fringe region. Due to the limitations of the K-means algorithm in capturing uncertain information, it cannot handle situations where an object might simultaneously belong to two or more clusters with similar proximity. To address this issue, the concept of Minimum Distance is introduced, which is used to determine the cluster with the highest similarity to each object.

Definition 5 *(Minimum Distance).* Let $U = \{x_1, x_2, \ldots, x_n, \ldots, x_N\}$ be a nonempty finite set of objects, and $C_{opt} = \{c_1^{opt}, c_2^{opt}, \ldots, c_k^{opt}, \ldots, c_K^{opt}\}$ be the optimal cluster centers of U, where K is the number of centers. For any $x \in U$, the minimum distance $d(x, c_{\min}^{opt})$ from object x to K centers is defined as:

$$d(x, c_{\min}^{opt}) = \min_{1 \le k \le K} dist(x, c_k^{opt}), \tag{18}$$

where $c_{\min}^{opt}$ represents the cluster center with the minimum distance from object x.

In addition, the Euclidean distance is used as the distance metric in this paper. To determine whether the data is uncertain, we introduce the concept of relative distances set based on the minimum distance. It is used to determine whether the difference between the minimum distance of object x and its distances to other optimal cluster centers is less than a threshold α. Based on this benchmark, we categorize data into two groups: deterministic data and uncertain data.

Definition 6 *(Relative Distances Set).* Let $U = \{x_1, x_2, \ldots, x_n, \ldots, x_N\}$ be a nonempty finite set of objects, and $C_{opt} = \{c_1^{opt}, c_2^{opt}, \ldots, c_k^{opt}, \ldots, c_K^{opt}\}$ be the optimal cluster centers of U, where K is the number of centers. For any $x \in U$, the relative distances set of object x is defined as:

$$Z(x) = \{j \mid |d(x, c_j^{opt}) - d(x, c_{\min}^{opt})| \le \alpha \text{ and } \min \ne j\}, \tag{19}$$

where $d(x, c_{\min}^{opt})$ is the minimum distance of object x, $d(x, c_j^{opt})$ is the distance between object x and the center of the jth cluster, and α is a predetermined parameter.

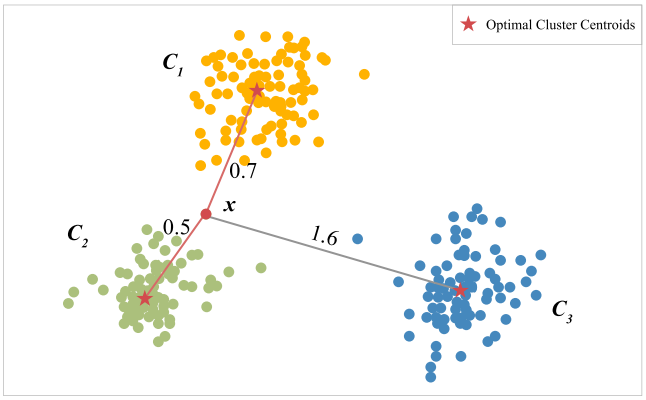


Fig. 7. Illustration of the Minimum Distance and the Relative Distances set.

When $Z(x) = \emptyset$, the object x belongs to only one cluster, which means that x is deterministic; when $Z(x) \neq \emptyset$, x may belong to multiple clusters, which means that x is uncertain, and the elements in $Z(x)$ correspond to the clusters that x may belong to.

To further clarify the concepts of the minimum distance and the relative distances set, we present an illustrative example, as shown in Fig. 7. Assume that the parameter α is equal to 0.2, and the distances between the object x and the three clusters C_1, C_2, and C_3 are $d(x, c_1^{opt}) = 0.5$, $d(x, c_2^{opt}) = 0.7$, and $d(x, c_3^{opt}) = 1.6$, respectively. In this scenario, the minimum distance of the object x is $d(x, c_2^{opt}) = 0.5$, and the relative distances set of x is $Z(x) = \{1\}$. Consequently, we can classify x as uncertain data, and conclude that x could potentially belong to either C_1 or C_2, but it is definitely excluded from C_3.

Definition 7 *(Upper Approximate Support).* Let $U = \{x_1, x_2, \ldots, x_n, \ldots, x_N\}$ be a nonempty finite set of objects, and $C_{opt} = \{c_1^{opt}, c_2^{opt}, \ldots, c_k^{opt}, \ldots, c_K^{opt}\}$ be the optimal cluster centers of U, where K is the number of centers. The upper approximate support R_k^{opt} of the kth cluster is defined as:

$$R_k^{opt} = Core(c_k^{opt}) \cup Fringe(c_k^{opt}). \tag{20}$$

The upper approximate support is the set of the upper approximate supports of all clusters, that is,

$$\mathbb{R}^{opt} = \{R_1^{opt}, R_2^{opt}, \ldots, R_k^{opt}, \ldots, R_K^{opt}\}. \tag{21}$$

For the two cases of the $Z(x)$, we assign object x to the upper approximate supports of the corresponding clusters according to the following rules.

$$\begin{cases} x \in R_{\min}^{opt} & if\ Z_x = \emptyset \\ x \in R_{\min}^{opt}\ and\ x \in R_j^{opt} (j \in Z(x)) & if\ Z_x \neq \emptyset \end{cases}, \tag{22}$$

where *min* corresponds to the cluster center with the minimum distance from object x.

Consider the extreme case where object x has the minimum distance from two or more cluster centers simultaneously. In this particular case, object x will be present in multiple approximate support sets for certain. Thus, one cluster center is randomly selected as the minimum distance for object x.

The upper approximate supports $\mathbb{R}^{opt}$ satisfy the following properties:

(1) $\forall R_k^{opt} \in \mathbb{R}^{opt},\ R_k^{opt} \neq \emptyset$

(2) $\bigcup_{i=1}^{K} (Core(c_k^{opt}) \cup Fringe(c_k^{opt})) = \mathbb{R}^{opt}$.

Property (1) ensures that the upper approximate support of each cluster is nonempty, which means that there is at least one object in each cluster; Property (2) ensures that all objects can be effectively partitioned by the upper approximate supports.

The object within $R_k^{opt} (1 \leq k \leq K)$ can be divided into the following two types:

$$\begin{cases} \text{Type I} = \{x \in R_k^{opt} \mid \forall j = 1, \ldots, K, j \neq k, x \notin R_j^{opt}\} \\ \text{Type II} = \{x \in R_k^{opt} \mid \exists j = 1, \ldots, K, j \neq k, x \in R_j^{opt}\} \end{cases}. \tag{23}$$

If an object x belongs to Type I, then x only exists in one upper approximate support, which means that x definitely belongs to a cluster. If an object x belongs to Type II, then x exists in multiple upper approximate supports, which means that x may belong to multiple clusters. Following the concept of three-way clustering, we assign objects belonging to Type II to the fringe region. As

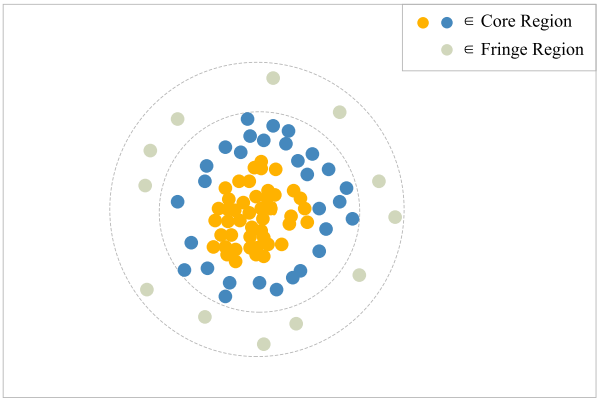


Fig. 8. Different similarities within the core region (Note: the orange data points in the core domain have a high similarity, while the blue data points have a low similarity).

for Type I, within the same cluster, there can also be variations in the strength of similarity among objects, as illustrated in Fig. 8. Taking into consideration the relationships within a cluster, we employ perturbation analysis and introduce the parameter β to further classify objects belonging to Type I into the core region and the fringe region. This method involves adding $\left|R_k^{opt}\right|$ identical objects to the upper approximate support of the kth cluster, where denotes the cardinality of R_k^{opt}. The new upper approximate support of the kth cluster, denoted as R_k^{opt*}, is then obtained. Using the mean-based Equation (4), we recalculate the center of R_k^{opt*}, and obtain a new center c_k^{opt*}. Then, we compare the distance between the new and old centers, denoted as $\left|c_k^{opt} - c_k^{opt*}\right|$, to determine whether object x belongs to the core region of the kth cluster. The core region is defined as follows.

Definition 8 *(Core Region).* Let $U = \{x_1, x_2, \ldots, x_n, \ldots, x_N\}$ be a nonempty finite set of objects, and $C_{opt} = \{c_1^{opt}, c_2^{opt}, \ldots, c_k^{opt}, \ldots, c_K^{opt}\}$ be the optimal cluster centers of U, where K is the number of centers. The core region of the kth cluster is defined as:

$$Core(c_k^{opt}) = \{x \mid x \in \text{Type I}, \ \left|c_k^{opt} - c_k^{opt*}\right| \leq \beta\}, \tag{24}$$

$$c_k^{opt*} = \frac{\sum_{x \in R_k^{opt*}} x}{\left|R_k^{opt*}\right|}, \tag{25}$$

$$R_k^{opt*} = R_k^{opt} \cup \{x^1, x^2, \ldots, x^{\left|R_k^{opt}\right|}\}, \tag{26}$$

where c_k^{opt} represents the current center, R_k^{opt*} represents the new upper approximate support of the kth cluster (calculated using Equation (26)), c_k^{opt*} represents the new center for R_k^{opt*} (calculated using Equation (25)), and β is a given parameter.

Definition 9 *(Fringe Region).* Let $U = \{x_1, x_2, \ldots, x_n, \ldots, x_N\}$ be a nonempty finite set of objects, and $C_{opt} = \{c_1^{opt}, c_2^{opt}, \ldots, c_k^{opt}, \ldots, c_K^{opt}\}$ be the optimal cluster centers of U, where K is the number of centers. The fringe region of the kth cluster is defined as:

$$Fringe(c_k^{opt}) = \{x \mid x \in \text{Type II}\} \cup (\text{Type I} \setminus Core(c_k^{opt})), \tag{27}$$

where c_k^{opt} represents the current center.

3.2. Algorithmic form of AHA-3WKM

Based on the concepts discussed above, this subsection provides a detailed description of the AHA-based three-way K-means clustering algorithm (AHA-3WKM). The parameter settings for AHA are as follows. First, the product of the dimension d of dataset and the number of clusters K is set to the population size, denoted as $nPop$. Second, the upper and lower limits, referred to as Up and Low, are defined based on the maximum and minimum values found within the dataset respectively. The pseudocode of AHA-3WKM is given in Algorithm 1, and its flowchart is illustrated in Fig. 9.

4. Experimental results and analysis

4.1. Clustering validity metric

Clustering performance measurement, also known as clustering validity metric, is a crucial process for evaluating the quality of clustering results. A good validity metric helps in comparing different clustering methods and analyzing whether one method is

Algorithm 1: AHA-3WKM

Input: $U = \{x_1, x_2, ..., x_N\}$, the number K of clusters, α, β, $Maxt$
Output: $CS^{opt} = \{(Co(c_1^{opt}), Fr(c_1^{opt})), (Co(c_2^{opt}), Fr(c_2^{opt})), ..., (Co(c_K^{opt}), Fr(c_K^{opt}))\}$

Initialization: The positions of $nPop$ hummingbirds are initialized using Equation (3), and their fitness values are calculated using fit_{AHA}. The visit table $VisitT$ is initialized using Equation (4).
Randomly select K objects as the initial cluster centers, denoted as $C = \{c_1, c_2, ..., c_k ..., c_K\}$.
for $t \leftarrow 1$ **to** $Maxt$ **do**
  for $i \leftarrow 1$ **to** $nPop$ **do**
    Randomly select a flight pattern and generate a direction change vector V according to Equations (5)-(7) ;
    Stochastically select either guided foraging or territorial foraging for the current hummingbird;
    if *guided foraging* **then**
      Perform Equation (8) and update the hummingbird's position and the optimal cluster center C^{opt} according to Equation (9). Update the visit table $VisitT$ based on the rules of guided foraging.
    else if *territorial foraging* **then**
      Perform Equation (10) and update the hummingbird's position and the optimal clustering centers C^{opt} according to Equation (9). Update the visit table $VisitT$ based on the rules of territorial foraging.
    end
    if $mod(t, 2nPop) == 0$ **then**
      Conduct the migration foraging, perform Equation (11), update the optimal cluster centers C^{opt} and adjust $VisitT$ according to the rules of migration foraging.
    end
  end
end
Obtain the optimal cluster centers $C^{opt} = \{c_1^{opt}, c_2^{opt}, ..., c_k^{opt} ..., c_K^{opt}\}$.
for $i \leftarrow 1$ **to** N **do**
  Calculate the minimum distance $d(x_i, c_{min}^{opt})$ of object x_i according to Equation (18);
  Determine the relative distances set $Z(x_i)$ according to Equation (19);
  if $Z(x_i) \neq \varnothing$ **then**
    $x_i \in R_{min}^{opt},\ x_i \in R_j^{opt},\ j \in Z(x_i)$;
  else
    $x_i \in R_{min}^{opt}$.
  end
end
for $i \leftarrow 1$ **to** K **do**
  Perform Equation (23), and determine sets Type I and Type II;
  Divided R_i into the core region and the fringe region according to Equations (24) and (27).
end

superior to another. Generally, clustering validity metrics are divided into two major categories: external index and internal index. This paper employs three external indices and two internal indices, aiming to assess the effectiveness of clustering algorithms.

(1) Accuracy (ACC)

ACC is a widely used external index, representing the proportion of correctly classified samples within the entire dataset. A higher value indicates a better clustering performance. It can be calculated according to Equation (28).

$$ACC = \frac{1}{N}\sum_{i=1}^{K} C_i, \tag{28}$$

where N is the number of samples; K is the number of clusters; C_i represents the number of samples correctly clustered into the ith cluster.

(2) Davies-Bouldin Index (DBI)

The Davies-Bouldin Index (DBI) is an internal index. It measures the separation between different clusters and the compactness within each cluster. A lower DBI value indicates higher intra-cluster similarity and lower inter-cluster similarity, suggesting better clustering performance. It can be calculated according to Equation (29).

$$DBI = \frac{1}{K}\sum_{i=1}^{K} \max_{j \neq i}\left(\frac{\overline{S}_i + \overline{S}_j}{M_{ij}}\right), \tag{29}$$

where K is the number of clusters; $\overline{S}_i$ is the average distance between samples in the ith cluster and the center of the ith cluster, representing the inner tightness of the cluster; M_{ij} represents the distance between the center of the ith cluster and that of the jth cluster, representing the degree of separation between clusters.

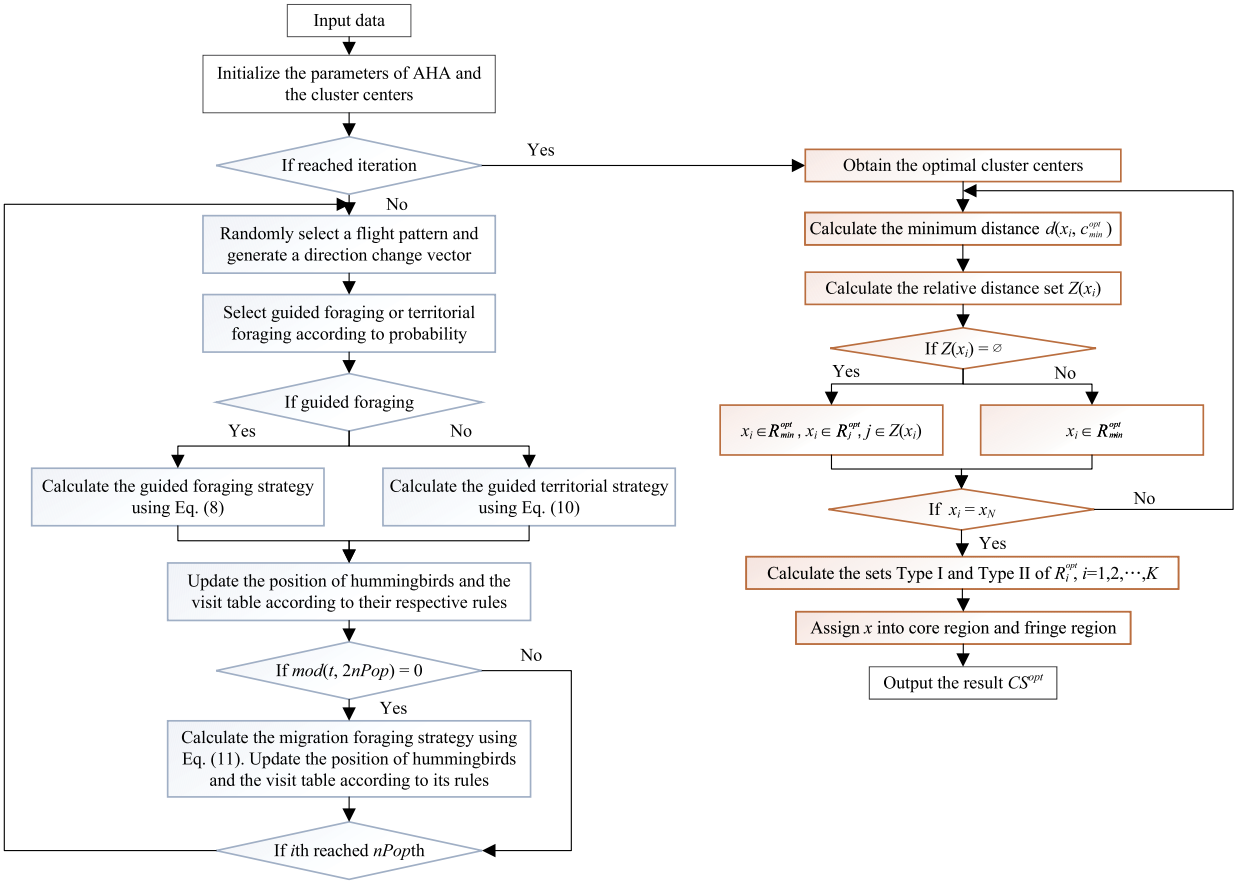


Fig. 9. The flowchart of AHA-3WKM. The blue part represents the process of finding the optimal cluster centers, and the red part represents the process of three-way clustering.

(3) Average Silhouette Index (AS)

The Average Silhouette Index (AS) is used to measure the consistency between clustering results and true labels. The calculation involves comparing pairwise matches between all data points, and adjusts for the influence of random matching. AS falls within the range of $[-1, 1]$, where a value closer to 1 indicates higher consistency between clustering results and true labels. It can be calculated according to Equation (30).

$$AS = \frac{1}{N}\sum_{i=1}^{N}\frac{b_i - a_i}{\max\{a_i, b_i\}}, \tag{30}$$

where N represents the number of objects, a_i represents the average distance between object i and other objects within the same cluster, and b_i represents the average distance between object i and the nearest cluster (other than its own).

(4) Adjusted Rand Index (ARI)

The Adjusted Rand Index (ARI) is a metric used to measure the consistency between clustering results and true labels. It considers all possible matching scenarios between clustering results and true labels and corrects for random matching. The value of ARI ranges from -1 to 1, with a value closer to 1 indicating higher consistency between clustering results and true labels. It can be calculated according to Equation (31).

$$RI = \frac{a+b}{\binom{N}{2}}, \tag{31}$$

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]}, \tag{32}$$

where N represents the number of objects, and $\binom{N}{2} = C_N^2 = \frac{N(N-1)}{2}$. Obviously, the range of RI is $[0, 1]$.

To calculate ARI, a contingency table is introduced, which displays the matching between clustering results and true labels. Rows in the contingency table represent the clusters from the clustering results, columns represent the categories from the true labels, and

Table 1
Description of 14 UCI datasets.

ID	Datasets	Number of samples	Dimensions	Number of classes
1	Iris	150	4	3
2	Wine	178	13	3
3	Seeds	210	7	3
4	WDBC	569	30	2
5	Glass	214	9	6
6	Breasttissue	106	9	6
7	Liver	345	6	2
8	Libras	360	90	15
9	CMC	1473	9	3
10	Ionosphere	351	34	2
11	Waveform	5000	21	3
12	Newthyroid	215	5	3
13	Balancescale	625	4	3
14	Vehicle	946	18	4

the values in the table represent the number of samples that belong to both the corresponding cluster and category. The adjusted ARI formula is defined as follows:

$$ARI = \frac{\sum_{ij}\binom{N_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] / \binom{N}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] / \binom{N}{2}}. \tag{33}$$

(5) Adjusted Mutual Information (AMI)

The Adjusted Mutual Information (AMI) measures the degree of information sharing between clustering results and true labels. Similar to ARI, AMI also corrects for random information and ranges from 0 to 1, with higher values indicating a better level of information sharing between clustering results and true labels. It can be calculated according to Equation (34).

$$AMI = \frac{MI(U,V) - E\{MI(U,V)\}}{F(H(U),H(V)) - E\{MI(U,V)\}}, \tag{34}$$

where $MI(U,V)$ represents the mutual information, $E\{MI(U,V)\}$ is the expected mutual information, $H(U)$ is the entropy of label U, and $H(V)$ is the entropy of label V. The function $F(H(U),H(V))$ can be chosen as the maximum function, geometric mean, or arithmetic mean. In this paper, we select the arithmetic mean as the function to adjust mutual information, and the calculation of AMI is adjusted as follows:

$$AMI = \frac{MI(U,V) - E\{MI(U,V)\}}{(H(U)+H(V))/2 - E\{MI(U,V)\}}. \tag{35}$$

4.2. Experimental environment and dataset

The experimental environment in this paper consists of an Intel Core i7 2.30 GHz CPU, 16 GB of RAM, a 512 GB hard drive, and the Windows 11 operating system. The programming was done using the Python language. To validate the effectiveness of AHA-3WKM, 14 UCI datasets were used in the comparative experiments. In order to make the data more general and representative, datasets with varying numbers of clusters, data points, and features were chosen. In addition, since the Euclidean distance used in this paper is not suitable for measuring distances between high-dimensional vectors, we do not consider high-dimensional datasets. The detailed information of the experimental datasets can be found in Table 1.

4.3. Experimental results of AHA-3WKM

In order to calculate ACC, DBI, AS, ARI, and AMI values, we selected the core regions in the clustering results. Smaller DBI values and larger ACC, AS, ARI, and AMI values indicate better clustering results. To highlight the improvements of the proposed algorithm on these performance metrics, it was compared with the other six algorithms: K-means, K-mediods, fuzzy c-means (FCM), TWKM [27], PSOK [31], and PTWC [28]. The detailed parameter settings for the above algorithms are shown in Table 2, where SSE represents the sum of squares due to error, N is the number of objects in the dataset, D is the dimension of the dataset, and K is the number of clusters. Note that the values of α and β for the proposed algorithm were determined based on the suggestions from Ref. [27] and extensive experiments. In addition, according to the suggestions from Ref. [19] on swarm intelligence algorithm and the results of numerous experiments, it was observed that AHA and PSO converge after 150 iterations. Therefore, the iteration number for both algorithms was set to 150.

To ensure that all features are on similar scales and avoid that any feature has a dominant impact on the model, each dataset was subjected to the min-max normalization in our experiment. In addition, each experiment was repeated 10 times, and the average

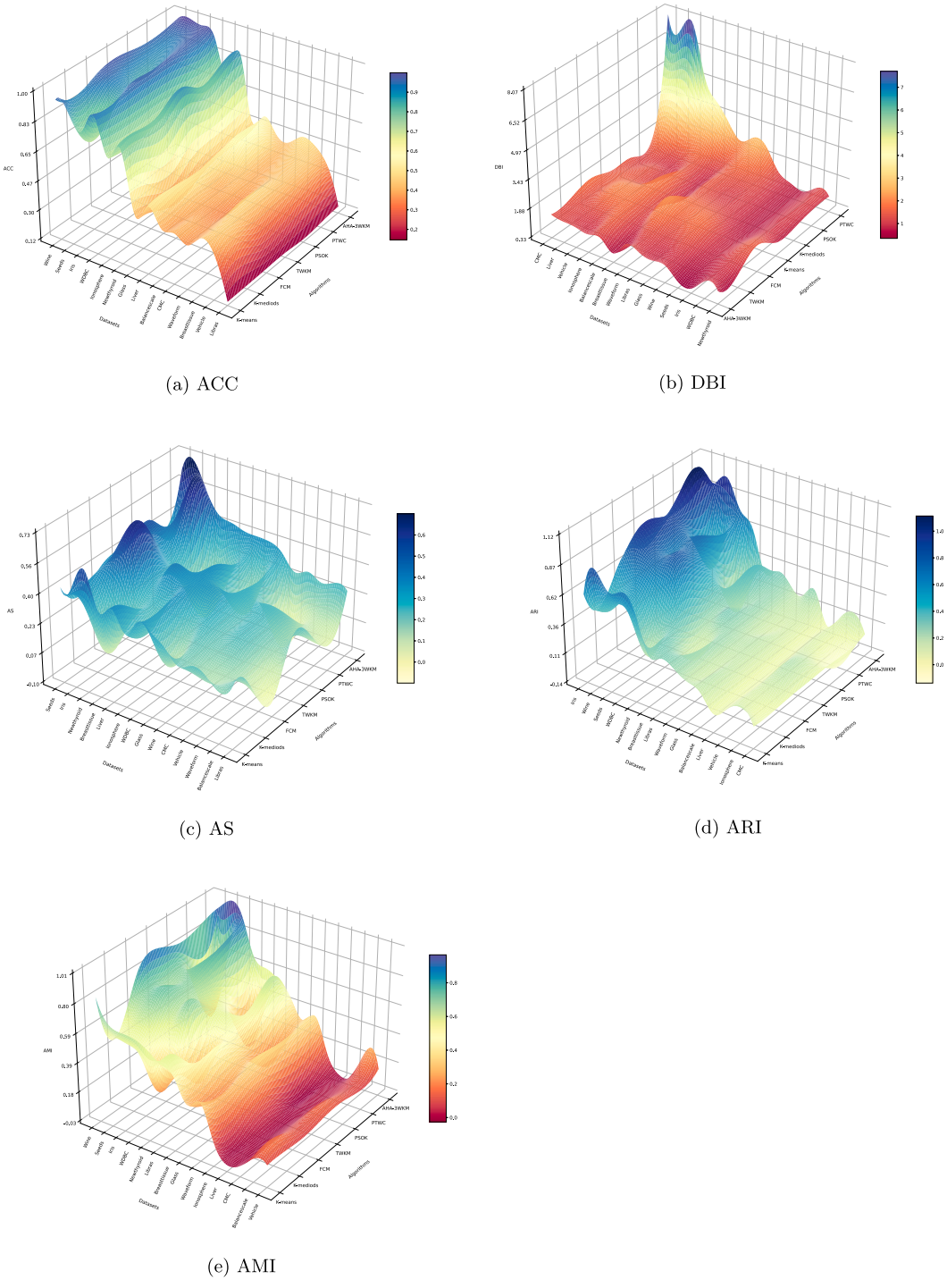
Table 2
Parameter settings of different algorithms.

Algorithms	Iterations	Fitness functions	Population size	Parameter Value 1	Parameter Value 2	Parameter Value 3
FCM	——	——	——	m = 2	error = 0.0005	max iterations = 1000
TWKM	——	——	——	α = 0.02	β = 0.00023×N	——
PSOK/PTWC	150	SSE	20	inertia factor = 0.9	learning factor 1 = 0.9	learning factor 2 = 0.5
AHA-3WKM	150	fit_{AHA}	$D \times K$	α = 0.02	β = 0.00023×N	——

Table 3
Comparison of experimental results on different datasets.

Datasets	Algorithm	ACC	DBI	AS	ARI	AMI	Datasets	Algorithm	ACC	DBI	AS	ARI	AMI
	K-means	0.8133	0.8277	0.4602	0.6048	0.6448		*K*-means	0.9438	1.3135	0.3001	0.8478	0.8316
	K-mediods	0.7467	0.9529	0.3370	0.4565	0.5436		*K*-mediods	0.8904	1.7921	0.1917	0.4573	0.4913
	FCM	0.8933	0.7746	0.4955	0.7287	0.7401		FCM	0.9494	1.3181	0.2993	0.8498	0.8318
Iris	TWKM	0.8859	0.7713	0.5074	0.6572	0.7058	Wine	TWKM	0.9697	1.1865	0.3300	0.9130	0.8961
	PSOK	0.9133	0.7881	0.4823	0.7709	0.7644		PSOK	0.9494	1.3086	0.3009	0.8537	0.8400
	PTWC	0.9488	0.7677	0.4316	0.7996	0.7984		PTWC	0.9506	1.4013	0.2892	0.8810	0.8687
	AHA-3WKM	**0.9811**	**0.4201**	**0.7123**	**0.9489**	**0.9322**		AHA-3WKM	**0.9775**	**1.1149**	**0.3510**	**0.9299**	**0.9169**
	K-means	0.8905	0.8766	0.4221	0.6991	0.6669		*K*-means	0.9297	1.1212	0.3874	0.7386	0.6350
	K-mediods	0.8238	1.1430	0.3115	0.5741	0.5537		*K*-mediods	0.9165	1.4063	0.3570	0.4663	0.4128
	FCM	0.9190	0.9312	0.4010	0.7723	0.7251		FCM	0.9315	1.1327	0.3817	0.7427	0.6288
Seeds	TWKM	0.8900	0.8727	0.4251	0.7031	0.6701	WDBC	TWKM	0.9327	1.1075	0.3931	0.7519	0.6506
	PSOK	0.8905	0.8759	0.4221	0.7049	0.6714		PSOK	0.9279	1.1363	0.3845	0.7302	0.6226
	PTWC	0.9136	0.9903	0.3595	0.3118	0.2922		PTWC	0.9011	1.2239	0.2998	0.6425	0.5490
	AHA-3WKM	**1.0000**	**0.4965**	**0.6459**	**1.0000**	**1.0000**		AHA-3WKM	**0.9357**	**1.0952**	**0.3985**	**0.7570**	**0.6575**
	K-means	0.4299	1.2092	0.3550	**0.1874**	**0.3135**		*K*-means	0.4434	0.9335	0.3609	0.2896	0.4721
	K-mediods	0.3855	1.8396	0.1169	0.1200	0.2253		*K*-mediods	0.4481	1.3656	0.1284	0.3376	0.4314
	FCM	0.3505	1.5432	0.2670	0.1614	0.2659		FCM	0.4717	1.0734	0.3028	0.2960	0.4785
Glass	TWKM	0.4258	1.1760	0.3791	0.1815	0.3063	Breasttissue	TWKM	0.4700	0.9143	0.3419	0.2644	0.4419
	PSOK	0.4678	1.5346	0.1702	0.1065	0.1672		PSOK	0.4497	1.1012	0.2297	0.3417	0.4531
	PTWC	0.4763	2.5633	0.0775	0.1326	0.2546		PTWC	0.4560	3.1514	0.0715	**0.3954**	**0.5293**
	AHA-3WKM	**0.5049**	**1.09638**	**0.39258**	0.17806	0.3104		AHA-3WKM	**0.4731**	**0.6511**	**0.4500**	0.2787	0.4463
	K-means	0.5113	1.3792	0.4076	-0.0039	-0.0023		*K*-means	0.1259	1.3414	0.2351	0.2983	0.5287
	K-mediods	0.5229	1.6176	0.3320	-0.0029	-0.0018		*K*-mediods	0.1278	1.9148	0.1180	0.1925	0.4003
	FCM	0.5026	1.6190	0.3013	-0.0049	-0.0015		FCM	0.1306	1.6761	-0.0326	0.0724	0.2163
Liver	TWKM	0.5130	1.3433	0.4170	-0.0046	-0.0021	Libras	TWKM	0.1324	1.3423	0.2459	0.3032	0.5322
	PSOK	0.5385	1.9469	0.0987	-0.0027	0.0005		PSOK	0.1222	1.7096	0.1707	0.2600	0.4711
	PTWC	0.5404	7.3083	-0.0004	**0.0046**	**0.0050**		PTWC	0.1338	2.8134	0.0273	0.3149	0.5328
	AHA-3WKM	**0.5459**	**1.3168**	**0.4288**	-0.0026	-0.0024		AHA-3WKM	**0.1447**	**1.2628**	**0.2593**	**0.3233**	**0.5469**
	K-means	0.4114	1.6072	0.2263	0.0251	0.0289		*K*-means	0.7108	1.5367	0.2949	0.1578	0.1185
	K-mediods	0.4070	1.8954	0.1882	0.0166	0.0126		*K*-mediods	0.6942	2.0845	0.2097	0.0474	0.0471
	FCM	0.3870	1.5369	0.2336	0.0242	0.0271		FCM	0.7094	1.5374	0.2945	0.1727	0.1280
CMC	TWKM	0.4258	1.5024	0.2570	0.0220	0.0249	Ionosphere	TWKM	0.7290	1.1515	0.4040	0.1747	0.1162
	PSOK	0.4291	1.5301	0.2557	0.0205	0.0233		PSOK	0.6942	1.5332	0.2165	0.0157	0.0338
	PTWC	0.4326	7.9799	-0.0292	0.0059	0.0279		PTWC	0.7070	5.2288	0.1637	-0.1031	0.0513
	AHA-3WKM	**0.4355**	**1.4493**	**0.2660**	**0.0331**	**0.0300**		AHA-3WKM	**0.7473**	**1.1458**	**0.4987**	**0.1905**	**0.1257**
	K-means	0.3703	1.4971	0.2329	0.2535	0.3639		*K*-means	0.7930	0.8474	0.5624	0.6283	0.5909
	K-mediods	0.3841	2.3006	0.1385	0.2528	0.2893		*K*-mediods	0.8047	1.5787	0.1207	0.2081	0.2843
	FCM	0.3705	1.5457	0.2251	0.2436	0.3299		FCM	0.8023	0.8872	0.5382	0.6927	0.6568
Waveform	TWKM	0.3860	1.5013	0.2327	0.2545	0.3690	Newthyroid	TWKM	0.8884	0.8462	0.5634	0.6346	0.5973
	PSOK	0.4214	1.4971	0.2330	0.2535	0.3639		PSOK	0.8225	1.0280	0.3202	0.4239	0.4671
	PTWC	0.4435	2.3737	0.0948	0.2546	0.3611		PTWC	0.8627	1.0041	0.5347	0.0090	0.1171
	AHA-3WKM	**0.4984**	**1.4232**	**0.2572**	**0.2558**	**0.3691**		AHA-3WKM	**0.8925**	**0.8459**	**0.5636**	**0.6359**	**0.5986**
	K-means	0.4104	1.7145	0.1703	0.1367	0.1178		*K*-means	0.3747	1.4414	0.2581	0.0820	0.1082
	K-mediods	0.4166	1.8012	0.1417	**0.1775**	**0.1489**		*K*-mediods	**0.4279**	2.0481	0.1406	0.0725	0.1028
	FCM	0.4413	2.0357	0.0815	0.1413	0.1089		FCM	0.3712	1.5268	0.2487	0.0745	0.0951
Balancescale	TWKM	0.4320	1.6959	0.1756	0.1581	0.1354	Vehicle	TWKM	0.3788	1.4021	0.2728	0.0856	0.1190
	PSOK	0.5482	1.6980	0.1609	0.1165	0.1082		PSOK	0.4009	1.4433	0.2564	0.0855	0.1213
	PTWC	**0.5529**	4.0935	0.0827	0.1022	0.0949		PTWC	0.4141	7.9369	-0.0104	0.0687	0.0937
	AHA-3WKM	0.4613	**1.6406**	**0.1842**	0.1578	0.1318		AHA-3WKM	0.3805	**1.3243**	**0.2839**	**0.1114**	**0.1640**

values were taken as the experimental results for the algorithms to compare their overall performance. The detailed experimental results for each dataset can be found in Table 3, with the best performance for each dataset highlighted in bold.



(a) ACC (b) DBI

(c) AS (d) ARI

(e) AMI

Fig. 10. A comparison of evaluation metrics on the 14 UCI datasets.

The experimental results in Table 3 demonstrate that the proposed algorithm outperforms other algorithms in 9 out of 14 datasets. Notably, it achieves the best values in terms of DBI and AS across all datasets. This result can be visually depicted in Fig. 10. In addition, AHA-3WKM correctly classifies all samples in the core region of the Seeds dataset. Although AHA-3WKM does not achieve the highest scores in all evaluation metrics for the Glass, Breasttissue, and Liver datasets, it performs well in terms of ACC, DBI, and AS. For the Vehicle dataset, although AHA-3WKM does not achieve the best performance in terms of ACC, it obtains the best performance in terms of the other four evaluation metrics. For the Balancescale dataset, while the performance of AHA-TWKM ranks fourth in terms of ACC, it still outperforms other algorithms in terms of DBI and AS.

Table 4
The running time (second) of different algorithms.

Datasets	*K*-means	*K*-mediods	FCM	TWKM	PSOK	PTWC	AHA-3WKM
Iris	0.0131	0.0143	0.0148	0.0941	1.4513	1.7204	8.7291
Wine	0.0116	0.0133	0.0139	0.0803	1.2035	1.2415	28.4824
Seeds	0.0149	0.0161	0.0169	0.1287	1.0870	1.5145	15.9862
WDBC	0.0163	0.0137	0.0172	0.1569	2.1202	2.7504	52.5227
Glass	0.0125	0.0122	0.0161	0.1112	1.5803	3.0076	37.3314
Breasttissue	0.0109	0.0083	0.0107	0.0812	1.7193	2.4740	36.6716
Liver	0.0114	0.0102	0.0122	0.0923	1.0091	1.2610	7.5809
Libras	0.0306	0.0273	0.0833	0.6527	17.7266	18.6240	210.9312
CMC	0.0464	0.0399	0.0873	0.5361	2.9159	3.1326	29.8079
Ionosphere	0.0228	0.0201	0.0211	0.1330	2.5788	2.8284	50.6678
Waveform	0.2818	0.2629	0.3169	3.0693	37.9876	39.1054	69.2494
Newthyroid	0.0087	0.0073	0.0186	0.0828	0.9199	1.2628	10.9537
Balancescale	0.0264	0.0236	0.1190	0.1650	1.7079	2.3224	12.5306
Vehicle	0.0200	0.0157	0.0340	0.4600	2.9750	3.0241	77.2584
Average	0.0377	0.0346	0.0559	0.4174	5.4987	6.0192	46.3360

In the proposed algorithm, the time complexity of AHA is determined by the size and dimensionality of the dataset, which means it may not have a computational advantage in terms of time complexity. The running time of different algorithms, i.e., the average time for each algorithm to repeat 10 times, can be found in Table 4. The time complexity of AHA-3WKM is $O(Tfn+Tnd+\frac{Td}{2})$, where T is the maximum number of iterations, f is the number of fitness evaluations, n is the population size, and d is the dimensionality. Fortunately, the impact of varying dataset sizes and dimensions on the performance of AHA-3WKM is relatively small.

Through the comparative analysis of experimental results, it can be observed that AHA-3WKM exhibits significant improvements on most datasets in terms of various clustering metrics. The improvement of AHA-TWKM can be attributed to the definition of the fitness function for AHA and the allocation of samples with uncertain information to the fringe regions, which are not taken into consideration when calculating evaluation metrics. Unlike traditional clustering algorithms, AHA-TWKM overcomes the problem related to local optima resulting from random selection of initial cluster centers, and it can successfully identify uncertain samples and make deferred decisions, which effectively reduces the risk of decision-making caused by blind clustering and improves the clustering results. In conclusion, this algorithm can effectively improve the clustering accuracy and demonstrate better clustering results.

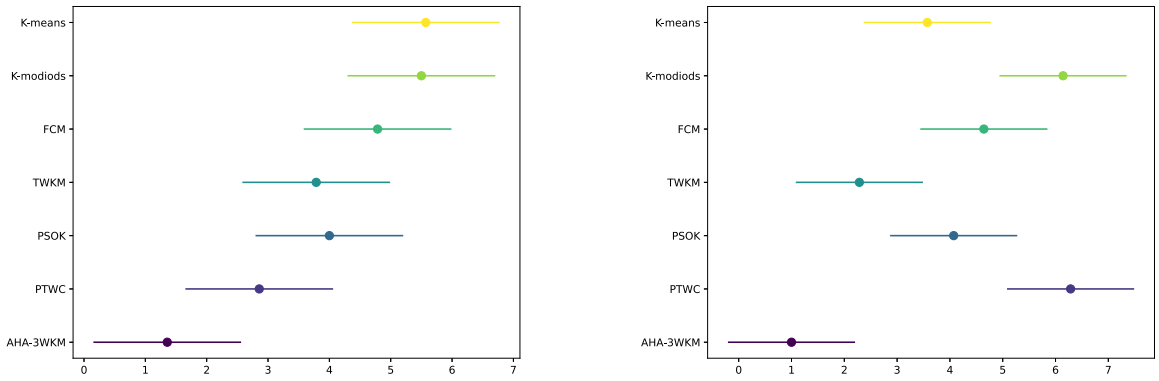
4.4. Statistical test analysis

In order to verify the advantages of the proposed algorithm, the Friedman test [32] and the Nemenyi post-hoc test [33] were used to determine the differences in performance among the five algorithms, with a focus on the ACC and DBI metrics.

First, the DBI values of each algorithm on each dataset were sorted in ascending order from good to bad and assigned ordinal values 1, 2, 3, ... If multiple algorithms have equal indicator values, their ordinal values are evenly divided. Second, the average ordinal value of each algorithm was calculated. Third, the results of the Friedman test were computed by τ_F and τ_{χ^2}, where follows the F-distribution with $(S-1)$ and $(S-1)(E-1)$ degrees of freedom (Note: the Friedman test is a non-parametric method used to assess the overall performance of S algorithms on E datasets). If the hypothesis that "all algorithms have the same performance" is rejected, then it indicates a significant difference in the performance of algorithms. In this case, further tests are needed to distinguish between the algorithms, and the Nemenyi post-hoc test was adopted in this paper. Fourth, the critical difference (CD) for the difference in average ordinal values was calculated by $CD = q_\alpha\sqrt{\frac{S(S+1)}{6E}}$, where q_α represents the critical value. If the difference in average ordinal values between two algorithms exceeds the CD, then the hypothesis that "the two algorithms have the same performance" is rejected with the corresponding level of confidence.

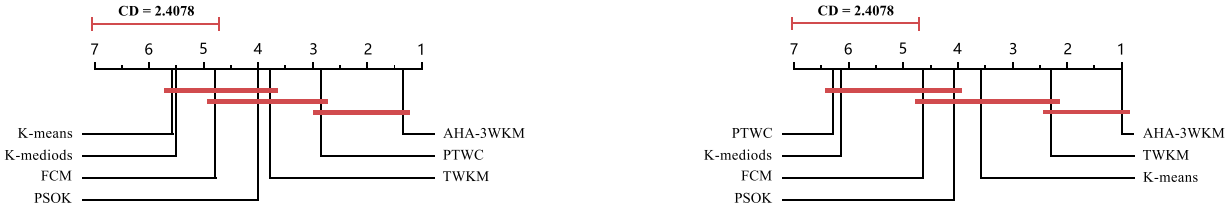
In our experiments, since $S=7$ and $E=14$, the data follows an F-distribution with 6 and 78 degrees of freedom. According to the Friedman test, when $\alpha=0.05$, if we consider ACC, then the τ_F value is 11.5168 which is greater than the critical value (i.e., 2.209); if we consider DBI, then the τ_F value is 51.5063 which is also greater than the critical value. Therefore, the null hypothesis that all algorithms have the same performance is rejected, and it can be concluded that there are significant differences in performance among all clustering algorithms. Fig. 11 visually represents these significant differences between algorithms.

When the significance level is 0.05, $q_{0.05}$ = 2.949. According to the Nemenyi post-hoc test, the CD is equal to 2.4078. The Nemenyi post-hoc test results under ACC and DBI are shown in Fig. 12. From Fig. 12, it can be seen that the performance of AHA-3WKM in terms of ACC and DBI ranks first compared to the other six algorithms, namely *K*-means, *K*-mediods, FCM, TWKM, PSOK, and PTWC. This indicates a significant difference of AHA-3WKM compared to the other six algorithms. Although the average ordinal values between AHA-3WKM and PTWC do not exceed the CD value in terms of ACC, and those between AHA-3WKM and TWKM do not exceed the CD value in terms of DBI, the actual rankings demonstrate that AHA-3WKM $\succ$ PTWC and AHA-3WKM $\succ$ TWKM. Consequently, from the statistical perspective, AHA-3WKM demonstrates a superior performance in the field of clustering.



(a) The Friedman test results under ACC (b) The Friedman test results under DBI

Fig. 11. The Friedman test results.



(a) The Nemenyi test results under ACC (b) The Nemenyi test results under DBI

Fig. 12. The Nemenyi test results.

5. Conclusions and future work

In this paper, we introduce a novel *K*-means method, AHA-3WKM, based on the artificial hummingbird algorithm and the three-way clustering. The proposed algorithm overcomes the limitations of traditional *K*-means algorithms in terms of sensitivity to initial cluster centers, local optimization problems, and difficulty in capturing data uncertainty. By incorporating AHA and defining a suitable fitness function, the flight and foraging strategies of hummingbirds are utilized to search for the optimal cluster centers in multiple iterations on the dataset. This optimization improves the stability and accuracy of the *K*-means algorithm. In addition, a three-way decision rule is used to divide the clustering results into three regions, effectively capturing information about data uncertainty and reducing the risk of decision-making. The effectiveness of AHA-3WKM was validated by evaluating five validity metrics, including ACC, DBI, AS, ARI, and AMI, on 14 UCI datasets and comparing them with the other six algorithms. The experimental results demonstrate the excellent performance and stability of the proposed algorithm. In future research, efforts will be made to enhance the efficiency and applicability AHA-3WKM in handling large datasets, due to its limitations when applied to high-dimensional data. In addition, effective approaches will be explored to tackle the challenges related to adaptive parameters, in order to improve the generalization ability of AHA-3WKM.

CRediT authorship contribution statement

Xiying Chen: Writing – original draft, Validation, Software, Methodology, Conceptualization. **Caihui Liu:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization. **Bowen Lin:** Software, Data curation. **Jianying Lai:** Software, Data curation. **Duoqian Miao:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

The research is supported by the National Natural Science Foundation of China under Grant Nos. 62166001, 61976158, Jiangxi Province Natural Science Foundation of China under Grant Nos. 20232ACB202013, Graduate Innovation Funding Program of Gannan Normal University under Grant No. YCX23A027.

References

[1] H. Zhao, Y. Wu, W. Deng, An interpretable dynamic inference system based on fuzzy broad learning, IEEE Trans. Instrum. Meas. 72 (2023) 2527412–2527423.
[2] Z. Chen, L. Fu, J. Yao, W. Guo, C. Plant, S. Wang, Learnable graph convolutional network and feature fusion for multi-view learning, Inf. Fusion 95 (2023) 109–119.
[3] Z. Fang, S. Du, X. Lin, J. Yang, S. Wang, Y. Shi, Dbo-net: differentiable bi-level optimization network for multi-view clustering, Inf. Sci. 626 (2023) 572–585.
[4] Y. Chen, Z. Wang, X. Bai, Fuzzy sparse subspace clustering for infrared image segmentation, IEEE Trans. Image Process. 32 (2023) 2132–2146.
[5] C.A. Åkerlund, A. Holst, N. Stocchetti, E.W. Steyerberg, D.K. Menon, A. Ercole, D.W. Nelson, Clustering identifies endotypes of traumatic brain injury in an intensive care cohort: a center-tbi study, Crit. Care 26 (1) (2022) 1–15.
[6] M. Ghiasabadi Farahani, J. Akbari Torkestani, M. Rahmani, Adaptive personalized recommender system using learning automata and items clustering, Inf. Sci. 106 (2022) 101978–101990.
[7] W. Deng, K. Li, H. Zhao, A flight arrival time prediction method based on cluster clustering-based modular with deep neural network, IEEE Trans. Intell. Transp. Syst. (2023).
[8] A.M. Ikotun, A.E. Ezugwu, L. Abualigah, B. Abuhaija, J. Heming, K-means clustering algorithms: a comprehensive review, variants analysis, and advances in the era of big data, Inf. Sci. 622 (2023) 178–210.
[9] P. Fränti, S. Sieranoja, How much can k-means be improved by using better initialization and repeats?, Pattern Recognit. 93 (2019) 95–112.
[10] P. Fränti, S. Sieranoja, K-means properties on six clustering benchmark datasets, Appl. Intell. 48 (2018) 4743–4759.
[11] L. Huang, H.-Y. Chao, C.-D. Wang, Multi-view intact space clustering, Pattern Recognit. 86 (2019) 344–353.
[12] J. Tang, G. Liu, Q. Pan, A review on representative swarm intelligence algorithms for solving optimization problems: applications and trends, IEEE/CAA J. Autom. Sin. 8 (10) (2021) 1627–1643.
[13] I.B. Saida, K. Nadjet, B. Omar, A new algorithm for data clustering based on cuckoo search optimization, in: Genetic and Evolutionary Computing: Proceedings of the Seventh International Conference on Genetic and Evolutionary Computing, ICGEC 2013, August 25-27, 2013-Prague, Czech Republic, Springer, 2014, pp. 55–64.
[14] R. Wang, Y. Zhou, S. Qiao, K. Huang, Flower pollination algorithm with bee pollinator for cluster analysis, Inf. Process. Lett. 116 (1) (2016) 1–14.
[15] J. Nayak, B. Naik, D. Kanungo, H. Behera, An improved swarm based hybrid k-means clustering for optimal cluster centers, in: Information Systems Design and Intelligent Applications: Proceedings of Second International Conference INDIA 2015, vol. 1, Springer, 2015, pp. 545–553.
[16] J. Nayak, B. Naik, H. Behera, Cluster analysis using firefly-based k-means algorithm: a combined approach, in: Computational Intelligence in Data Mining: Proceedings of the International Conference on CIDM, 10-11 December 2016, Springer, 2017, pp. 55–64.
[17] Y. Li, X. Zhou, J. Gu, K. Guo, W. Deng, A novel k-means clustering method for locating urban hotspots based on hybrid heuristic initialization, Appl. Sci. 12 (16) (2022) 8047–8073.
[18] W. Zhao, L. Wang, S. Mirjalili, Artificial hummingbird algorithm: a new bio-inspired optimizer with its engineering applications, Comput. Methods Appl. Mech. Eng. 388 (2022) 114194–114239.
[19] S. SI-MA, H. Liu, H. Zhan, G. Guo, C. Yu, P. Hu, Swarm intelligence algorithms evaluation, arXiv e-prints, 2023.
[20] S. Zhao, D. Wang, Elite-ordinary synergistic particle swarm optimization, Inf. Sci. 609 (2022) 1567–1587.
[21] B.S. Yildiz, P. Mehta, S.M. Sait, N. Panagant, S. Kumar, A.R. Yildiz, A new hybrid artificial hummingbird-simulated annealing algorithm to solve constrained mechanical engineering problems, Mater. Test. 64 (7) (2022) 1043–1050.
[22] D. Yousri, H.E. Farag, H. Zeineldin, E.F. El-Saadany, Integrated model for optimal energy management and demand response of microgrids considering hybrid hydrogen-battery storage systems, Energy Convers. Manag. 280 (2023) 116809–116827.
[23] Y. Yao, The dao of three-way decision and three-world thinking, Int. J. Approx. Reason. (2023) 109032–109053.
[24] Y. Yao, The superiority of three-way decisions in probabilistic rough set models, Inf. Sci. 181 (6) (2011) 1080–1096.
[25] Y. Yao, The geometry of three-way decision, Appl. Intell. 51 (9) (2021) 6298–6325.
[26] Y. Hong, Three-way cluster analysis, Peak Data Sci. 5 (1) (2016) 31–35 (in Chinese with English Abstract).
[27] P. Wang, H. Shi, X. Yang, J. Mi, Three-way k-means: integrating k-means and three-way decision, Int. J. Mach. Learn. Cybern. 10 (2019) 2767–2777.
[28] C.R. Gao Yanlong, Wan Renxia, A three-way clustering algorithm based on particle swarm optimization, J. Fuzhou Univ. 50 (3) (2022) 301–307 (in Chinese with English Abstract).
[29] Q. Guo, Z. Yin, P. Wang, An improved three-way k-means algorithm by optimizing cluster centers, Symmetry 14 (9) (2022) 1821–1838.
[30] A.E. Ezugwu, A.M. Ikotun, O.O. Oyelade, L. Abualigah, J.O. Agushaka, C.I. Eke, A.A. Akinyelu, A comprehensive survey of clustering algorithms: state-of-the-art machine learning applications, taxonomy, challenges, and future research prospects, Eng. Appl. Artif. Intell. 110 (2022) 104743–104786.
[31] H. Zhang, Q. Peng, Pso and k-means-based semantic segmentation toward agricultural products, Future Gener. Comput. Syst. 126 (2022) 82–87.
[32] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1) (1940) 86–92.
[33] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.